Análisis Forense de Memoria: *Malware* y Evidencia Oculta

Ana Haydée Di Iorio, Gonzalo Ruiz de Angeli, Juan Ignacio Alberdi, Hugo Curti, Fernando Greco, Ariel Podestá, Martin Castellote, Bruno Constanzo, Juan Iturriaga, Santiago Trigo

Universidad FASTA
{diana, arieluf, bconstanzo}@ufasta.edu.ar

Resumen. En este trabajo se presenta el tema de análisis forense de memoria principal, como una herramienta valiosa para las investigaciones judiciales. En particular, se expone la capacidad de las técnicas de análisis de memoria para encontrar tanto estructuras propias del sistema operativo que solamente residen en memoria, como estructuras ocultas que indicarían la presencia de malware en el equipo. Este tipo de información puede ser de gran ayuda para casos complejos, en donde el análisis informático forense clásico no resulta suficiente.

1 Introducción

En la investigación de ciertos casos penales, puede ser crítica la información contenida en la memoria principal de un dispositivo computacional al momento del secuestro del equipo en un allanamiento o escena del crimen[1, 2].

A partir de la preservación de la información contenida en memoria principal, es posible luego realizar un análisis minucioso y exhaustivo de los procesos en ejecución al momento del secuestro para identificar si las computadoras estuvieron involucradas en actividades ilícitas, ya sea hacia o desde éstas, con el objetivo final de colaborar en una causa judicial. Ciertos ataques no dejan rastros en el disco rígido, por lo que sólo será posible encontrar indicios del hecho mediante el análisis de la memoria principal. Llevando a cabo el análisis forense de memoria es posible identificar qué procesos estuvieron ejecutando y desde cuándo, entre otros datos importantes, que puedan derivar en información relevante para la investigación.

El análisis de memoria es una disciplina nueva en la informática forense, que se ha comenzó a desarrollar hace poco más de diez años, pero con especial énfasis en los últimos cinco. Si bien múltiples herramientas proveen la capacidad de capturar un volcado de memoria, es menor la cantidad que permiten realizar el análisis forense sobre los mismos. A modo de ejemplo, en el ámbito comercial se encuentran Mandiant Memoryze y MoonSols Windows Memory Toolkit, mientras que en el ám-

bito Open Source se existen Volatility[1] y Rekall. Ha habido otros productos anteriores, tanto comerciales como libres, pero debido a los desafíos y la constante actualización que requiere el desarrollo y mantenimiento de este tipo de herramientas, pocos proyectos han perdurado en el tiempo.

La necesidad de profundizar en el conocimiento y aplicación de este tipo de técnicas fue detectada en el año 2012 por el Grupo de Investigación de Sistemas Operativos e Informática Forense de la Universidad FASTA y se propuso un Proyecto Final de Graduación en la temática (BIP-M), abordado por Gonzalo Ruiz de Angeli y Juan Ignacio Alberdi. A finales de 2015 se presenta dicho proyecto, y satisfechos con los resultados del mismo, se continuó con la investigación en esta área temática como un proyecto de investigación completo, con la participación de todo el Grupo de Investigación. Este nuevo proyecto, DIMA, busca canalizar el conocimiento adquirido previamente para aplicarlo al análisis de malware y la creación de indicadores que ayuden a su detección.

El *malware* es software malintencionado que pretende ser indetectable y hacer uso del sistema operativo con el más alto nivel de privilegios, para poder llevar adelante su objetivo. Es por esto que esta clase de software se relaciona con procesos, muchas veces ocultos, en la memoria. Algunos tipos de malware[3], según sus características, puede ser:

- *Backdoor*: código malicioso que se instala y brinda acceso a un atacante, permitiendo que ejecute comandos sobre el sistema.
- *Botnet*: al igual que un *backdoor*, permite el acceso a un sistema, pero se diferencia en que una botnet se compone de muchas computadoras comprometidas, que reciben el mismo comando desde un servidor de *command-and-control* (C&C, comando y control).
- *Downloader*: código malicioso que realiza la descarga e instalación de uno o más componentes en un ataque complejo.
- *Launcher*: código malicioso que se ocupa de garantizar la ejecución de otros componentes en un *malware* complejo.
- *Scareware*: *malware* cuyo propósito principal es inducir miedo en el usuario (con la finalidad de lograr un objetivo, por ejemplo, la descarga e instalación de otro *malware*).
- Gusano o virus: código malicioso que puede copiarse e infectar otras computadoras.

[1] Los autores de este trabajo han contribuido al desarrollo de Volatility, y sus aportes fueron incorporados a partir de la versión 2.5 de este *framework*.
https://github.com/volatilityfoundation/volatility/pull/88
https://github.com/volatilityfoundation/volatility/commit/a52c979df52efcddaf4ea85fa5e7cb11bc258661#diff-ec6b233a4376469c2d28e2f18a4a7f98

- Caballos de Troya (o troyanos): programas que muestra una interfaz o funcionalidad, pero en realidad esconden código malicioso que se ejecuta de manera desapercibida.
- *Spyware*: *malware* que recopila información del usuario y la computadora infectada para luego enviarla a un atacante.
- *Rootkits*: código malicioso diseñado para infectar el sistema operativo desde el más bajo nivel para ocultarse a sí mismo, y a otros componentes de *malware*.
- *Ransomware*: una variante moderna del *scareware*, este tipo de *malware* cifra archivos de la computadora y luego exige el pago de una recompensa a cambio de la clave de cifrado para recuperar la información perdida.

Si bien estas categorías son útiles para organizar el estudio, en general, el *malware* moderno es complejo y puede estar compuesto de varias partes que cumplen con una de estas clasificaciones, lo que resulta en que un sólo *malware* abarca varias categorías.

Ante esta situación, el análisis forense de la memoria principal se vuelve una herramienta imprescindible para los casos en donde pueda haber evidencia digital[2] relacionada con un *malware*, debido a que puede haber información de gran importancia para la investigación, oculta tras un velo levantado por los atacantes para esconder sus acciones.

2 Análisis forense de la memoria principal

El análisis forense en memoria consiste en el procesamiento e interpretación de datos provenientes de la memoria principal[3] con el fin de obtener información de interés. Este estudio es posible llevarlo a cabo "en vivo", es decir, analizando la memoria en un momento determinado mientras el sistema está en funcionamiento; pero también existe otra manera: a través de la obtención del volcado de memoria (*memory dump*) donde se copia en un archivo el contenido de toda la memoria en un momento determinado. De esta manera, es posible realizar el análisis con una réplica exacta o “copia forense” de la memoria del dispositivo en cuestión. Este último método puede resultar el indicado en muchos casos, dado que es menos intrusivo: no genera actividad en memoria que pueda contaminar la evidencia o la contaminación es mínima, y a su vez, al estar almacenado en un archivo, permite realizar sucesivos análisis a futuro sin modificar ni perder la evidencia.

---

[2] Se define evidencia digital como la información procesada por un sistema informático, almacenada en un soporte digital, que resulta de interés para una investigación o causa judicial [6].

[3] La memoria principal de una computadora es aquella que se encuentra conectada directamente al procesador a través del bus interno. Es normalmente conocida como memoria RAM o memoria volátil.

Independientemente de la forma de trabajo, ya sea "en vivo" o con volcado de memoria, es necesario conocer las estructuras que se pueden encontrar. Estas estructuras representan los diferentes artefactos de un sistema operativo: procesos, hilos (*threads*), módulos, conexiones, *sockets*, controladores (*drivers*), entradas de registro, y otros. Dependiendo del Sistema Operativo y de la arquitectura de la computadora, las estructuras presentan diferentes diseños[4, 5]. En la mayoría de los casos esta información está disponible únicamente en memoria principal.

Teniendo en cuenta estas consideraciones, es posible analizar la memoria con el fin de obtener información relevante que pueda luego convertirse en evidencia digital. En el proceso de análisis hay que tener en cuenta que pueden existir objetos que se encuentren escondidos: estructuras ocultas y *malware*.

En la Figura 1 se puede observar un ejemplo de esquema reducido de un procedimiento completo de análisis forense en memoria principal.

Fig. 1. Esquema reducido del proceso de análisis forense de memoria principal

En el proceso pueden reconocerse las tareas del analista forense remarcadas en recuadros color naranja, y en los recuadros azules las tareas que se realizan con una o varias herramientas de análisis de memoria. El reconocimiento de estructuras consiste en identificar los distintos artefactos del sistema operativo que pueden encontrarse dentro de la memoria explorando el volcado que se generó en la etapa previa.

El paso siguiente consiste en relacionar los distintos artefactos identificados. Por ejemplo: relacionar los procesos con las conexiones TCP, o procesos con módulos; y así poder establecer vínculos entre los datos recolectados que puedan resultar en información útil para el experto forense.

El análisis automático consiste en la obtención automatizada de información a partir de los datos identificados previamente, como puede ser la detección de posibles procesos ocultos o adulteraciones al Sistema Operativo, y demás información que pueda posteriormente ser analizada por el informático forense. Para la realización de

esta tarea se requiere software específico que sea capaz de interpretar las estructuras de un determinado kernel de un Sistema Operativo.

3 Framework BIP-M: Búsqueda de Información de Procesos en Memoria

BIP-M (Búsqueda de Información de Procesos en Memoria) es un *framework* para el análisis forense de memoria que surge como una herramienta alternativa en un contexto donde aún es necesario dar respuesta a cuestiones de análisis forense en memoria principal. El objetivo de este proyecto, surgido como proyecto final de graduación de la carrera de Ingeniería Informática de la Universidad FASTA y asociado al Grupo de Investigación de Sistemas Operativos e Informática Forense fue desarrollar un *framework* que permita el análisis de volcados de memoria del tipo *CrashDump* de sistemas operativos Microsoft Windows 7, tanto para la versión x86 como para x64.

En el año 2012, cuando comienza el proyecto, las herramientas de análisis de memoria existentes dejaban mucho que desear: la mayoría solo permitía el análisis de memoria de Windows XP, el soporte de las arquitecturas de 64 bits era pobre, y los proyectos Open Source existentes en ese momento se encontraban prácticamente abandonados. El Proyecto BIP-M, ambicioso desde su concepción, fue una forma de atacar este nicho carente, identificado desde el Grupo de Investigación, y también adquirir conocimiento en profundidad sobre los mecanismos de los Sistemas Operativos.

A partir del diseño de su arquitectura, es posible extender su funcionalidad para el análisis de otros sistemas operativos, así como también de otros formatos de volcado de memoria. A su vez, permite la salida de los resultados del análisis por pantalla y el almacenamiento de los mismos en una base de datos (actualmente implementado en MySQL), permitiendo extender el método de persistencia de los resultados sobre el soporte que se requiera gracias a su diseño flexible. Esta persistencia en base de datos, ofrece la posibilidad de realizar un posterior análisis de relaciones basado en consultas SQL, así como también, obtener en segundos los resultados de análisis ya realizados sin incurrir, nuevamente, en el tiempo que requiere buscar estructuras en un volcado de memoria de varios gigabytes, por ejemplo, utilizando el método *pool-tag scanning*[4] [7, 8].

[4] *Pool-tag scanning* es una técnica de búsqueda de estructuras en memoria basada en una etiqueta (*tag*). Por ejemplo, en el caso de estructuras de procesos, mediante esta técnica se realiza la búsqueda en todo el volcado de memoria de la etiqueta “Proã”. Cada ocurrencia de dicha etiqueta, puede indicar la presencia de un proceso en esa ubicación del archivo de volcado de memoria. Esta técnica se explica en detalle más adelante.

4 Captura del volcado de memoria

El contenido en memoria cambia constantemente, de un instante a otro. Incluso durante el proceso de adquisición del volcado de memoria es necesario utilizar alguna herramienta que es ejecutada en el equipo y que, en consecuencia, utiliza la misma memoria a capturar y analizar. Es por este motivo que es importante entender que la elección de la herramienta a utilizar debe tener en cuenta algunas cuestiones:

1. Identificación del/los procesos propios de la herramienta.
2. Espacio que ocupa el/los procesos de la herramienta al momento de ejecutarla.
3. Tipo de volcado que genera la herramienta: formato e información que vuelca desde la memoria al archivo (Por ejemplo, *Crash Dump* vs *Raw Dump*, posibilidad de volcar el contenido del *page file*, etc).

Como cualquier escena de un crimen, es de suma importancia contaminar ésta lo menos posible, motivo por el cual, teniendo en consideración todo lo expuesto previamente, es prioridad generar el volcado de memoria antes de avanzar con cualquier otro análisis o tarea sobre el equipo.

5 Estructuras en memoria: artefactos ocultos y *malware*

Los artefactos que se pueden encontrar en memoria son variados y, por consecuencia, también las estructuras que los representan. En memoria es posible encontrar, entre otras cosas:

- Procesos: activos, terminados y ocultos
- Hilos (*Threads*)
- Módulos y DLLs[5]
- Archivos abiertos por procesos
- Conexiones
- *Sockets*
- Contenido cifrado
- Claves asociadas a cuentas de usuario
- Entradas de registro
- Controladores (*Drivers*)
- Información relacionada con las cuentas de usuario y privilegios de acceso asociados a un proceso

[5] Los módulos y *Dynamic Link Libraries* (DLLs) son mecanismos que reúnen funcionalidades comunes a múltiples programas en un solo ejecutable cargado en memoria, ya sea parte del Sistema Operativo o desarrollados por terceros.

- Listado de usuarios conectados al equipo, local o remotamente.

Cada uno de estos artefactos está representado por una (o más) estructura de datos, las cuales poseen diversos formatos y tipos de datos, que pueden diferir de un sistema operativo a otro, dependiendo a su vez de la arquitectura de 32 bits o 64 bits. El conocimiento y estudio de estas estructuras es la base o punto de partida para llevar adelante un análisis forense de los objetos que podemos encontrar en la memoria. Antes de avanzar en un análisis de estas características, debemos entender cada uno de estos artefactos y cómo se relacionan entre sí para lograr obtener información coherente a partir de los datos identificados.

El análisis forense en memoria es fundamental para la búsqueda de malware y/o datos que se encuentran ocultos a simple vista. Conociendo qué estructuras se pueden encontrar, qué datos almacenan, cómo es su disposición en la memoria y cómo se relacionan con artefactos del mismo tipo o de otro, se está en condiciones de llevar el análisis a otro nivel.

Particularmente, el *software* malintencionado intenta ejecutarse oculto de la percepción del usuario. En una situación normal, los procesos activos se organización en una lista, mantenida en memoria por el sistema operativo, y poseen un puntero a su predecesor y a su sucesor. En el caso del *malware*, los procesos se desasocian de dicha lista para no ser detectados fácilmente, y contaminan alguna interrupción o llamada del sistema para poder tomar el control y ejecutarse de manera sigilosa.

La Figura 2 muestra gráficamente un proceso oculto en la memoria:

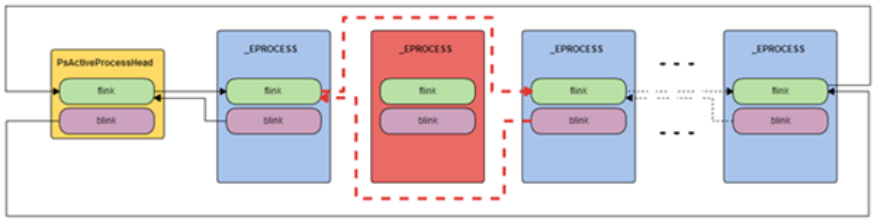

Fig. 2. Lista circular doblemente enlazada de procesos en memoria. Se puede apreciar un proceso oculto, desasociado de la misma.

Aun así, es posible encontrar indicios que permitan detectar este tipo de *software*. Gracias a los antes mencionados *pools* de memoria, es posible realizar una búsqueda de estructuras a través de la técnica *pool-tag scanning*. Ésta consiste en buscar en todo el *dump*, las ocurrencias del *tag* de la estructura a buscar: en el caso de los procesos, el *tag* es “Proã”. Una vez detectado el *tag*, es posible construir la estructura del proceso, y así sucesivamente con cada ocurrencia. Cada uno de estos hallazgos formará parte de una colección procesos que, por último, se deberá comparar con la lista de procesos activos. Aquellos que no estén en esta última, serán los procesos sospechosos que requerirán un mayor análisis. Se debe tener en cuenta respecto a esta técnica que es posible encontrar en el camino varios falsos positivos, los cuales

deberán ser descartados posteriormente por el perito forense a medida que realiza el análisis.

6 Análisis forense de la memoria principal en investigaciones judiciales

El objetivo del análisis forense en el marco de una actuación judicial es el hallazgo de evidencia digital, de indicios que permitan al investigador responder algunas de las preguntas básicas respecto a un hecho: ¿Cómo sucedió? ¿Cuándo sucedió? ¿Quienes participaron?, entre otras. Particularmente, en la memoria principal se almacenan datos que pueden ser de gran relevancia, y convertirse así en evidencia digital. Por su carácter altamente volátil, los datos en la memoria principal deben capturarse antes de cualquier otra actividad, como, por ejemplo, antes de la adquisición del contenido del disco rígido

Conociendo las particularidades y generalidades del análisis forense en la memoria principal, resulta necesario entender en qué situaciones puede llegar a ser de mayor utilidad o no la búsqueda de uno u otro artefacto en la memoria principal y sus relaciones.

La búsqueda de procesos puede parecer el eje del análisis forense en la memoria principal, pero también existen artefactos en dicha memoria que no necesariamente representan a los propios procesos y que pueden llegar a contener información de gran utilidad para el analista forense. El análisis de la memoria principal puede ayudar a inferir el uso que se le da al equipo, o detectar indicios que soporten una hipótesis particular.

Uno de los principales motivos por los que puede ser necesario realizar la adquisición del contenido de la memoria principal de un equipo encendido al momento de un allanamiento, es descartar la presencia de *malware*, el cual puede ocasionar que cierta acción parezca realizada por un usuario del equipo a analizar, cuando en realidad es realizada por otro mediante el uso indebido de dicho equipo a distancia. Si bien un *malware* puede detectarse mediante el hallazgo en el disco rígido del equipo de los archivos que lo componen, estos podrían encontrarse cifrados u ocultos dentro de un programa del sistema operativo comprometido. Por esta razón, es más fácil, y también de mayor utilidad, identificarlo en memoria a través del proceso y las conexiones que establece. Además, de esta manera podemos asegurar que se encontraba en ejecución momento de la captura del volcado de memoria, y luego verificar su tiempo de ejecución para enmarcarlo al momento de ocurrencia del hecho.

Este tipo de análisis, también, permite obtener las claves que estuvieran cargadas en la memoria principal y que dan indicios de la actividad del usuario. Otra gran utilidad del volcado de memoria es el acceso a las claves de cifrado que podrían ser requeridas en el futuro análisis del disco rígido del equipo.

Para ir más en detalle, a continuación, se realiza un listado de motivaciones para realizar análisis forense de la memoria principal y qué elementos serían objeto de análisis en cada caso:

- Búsqueda de información sobre el comportamiento del usuario: registro de eventos, artefactos relacionados con el sistema de archivos, como archivos, registro de eventos, entradas de registro, conexiones y *sockets*, fragmentos legibles.
- Búsqueda de *Malware*: procesos ocultos, librerías, conexiones, contenido cifrado, entradas de registro: antes de comenzar con el análisis de procesos, es imprescindible conocer los procesos críticos del sistema operativo objeto de estudio en el volcado de memoria, a fin de identificar procesos sospechosos partiendo desde lo más básico, como puede ser similitud en nombres de procesos de sistema variando algunas letras en su nombre Por ejemplo, csrss.exe es un archivo de sistema de Windows 7 que se carga durante el inicio y se encuentra en el directorio \System32 y debe aparecer una única instancia en la lista de procesos, con lo cual si se observa el nombre duplicado, fuera del directorio \System32 o con algún nombre similar como csrsss.exe, estamos en presencia de un potencial *malware*. Si bien difícilmente el analista se encuentre con el caso de la presencia de un proceso llamado malware.exe, no se debe descartar el análisis de procesos con nombres que despiertan sospecha, por más triviales que parezcan.

  A su vez, es fundamental tener conocimiento de otros programas de uso común como editores de texto, herramientas de ofimática, navegadores, compresores, antivirus, etc.; que pueden existir en equipos a fin de poner foco en procesos que resulten inusuales y sean objeto de análisis.

  Basado en la búsqueda de procesos, a partir de ellos es posible descubrir muchos datos que pueden llevarnos a relacionar el mismo con actividades ilícitas, con este, los artefactos relacionados como librerías (como DLLs), conexiones, usuario que ejecuta el proceso, contexto de seguridad, nivel de privilegios, etc.
- Búsqueda del contexto de seguridad de cada proceso: puede identificarse el contexto de seguridad obtenido, esto es el nivel de privilegios con el cual ejecuta (en Windows, la estructura que persiste esta información en la memoria principal para cada proceso es _TOKEN), como puede ser modo usuario o de sistema operativo (*kernel*), su proceso padre, y otros elementos mencionados previamente que permitan realizar un análisis y establecer conclusiones, arribando con esta información a la confirmación de existencia o no de software malicioso involucrado en actividades ilícitas.
- Búsqueda de información que no se encuentra en el disco y que se sospecha se quiso ocultar: búsqueda de contenido cifrado, búsqueda de contenido “legible” (se pueden encontrar fragmentos de texto de un archivo, un docu-

mento o un correo electrónico que ha sido leído o escrito), búsqueda de archivos.

- Búsqueda de información que indique comunicación con otros equipos: conexiones y sockets: dentro de los datos de conexiones, ya sea TCP, UDP y *sockets*[6], es posible vincular las mismas con los identificadores de procesos que las establecieron y así, llegar hasta los mismos, pudiendo verificar si estos resultan sospechosos, por ejemplo, si fueran procesos ocultos. Así se puede relacionar dichos procesos con conexiones remotas, y a partir de esta información obtenida localmente, es factible extender el análisis más allá del propio equipo elemento de estudio, indagando sobre una posible ubicación del equipo remoto, entre otras cosas, a fin de avanzar con mayor información vinculante con actividades ilícitas.
- Búsqueda de información que indique la utilización de algún dispositivo periférico en el equipo: búsqueda de drivers.
- Búsqueda de presencia de algún software específico: procesos, entradas de registro, módulos.

Sin embargo, el análisis de los datos en memoria no se circunscribe a la búsqueda de estructuras específicas y a la relación entre ellas según su diseño, sino que también es importante extender dicho análisis a la búsqueda de datos que puedan ser relevantes y que no formen parte de ninguna estructura en particular. Por ejemplo, en la En la Figura 3 se puede ver parte del contenido de lo que ha sido una página web abierta en el equipo desde donde se realizó el volcado de la memoria principal:

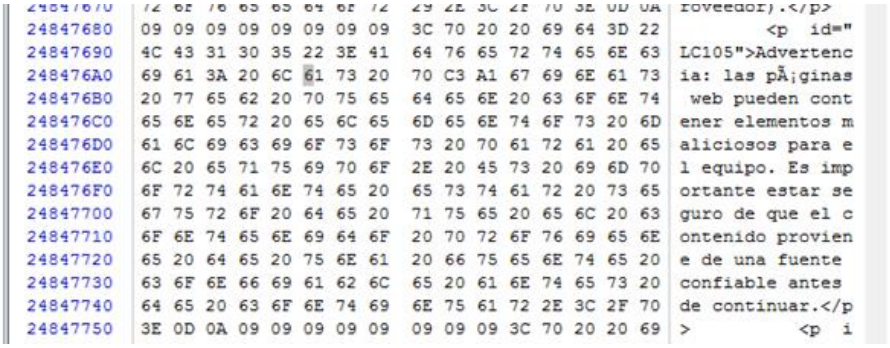

Fig. 3. Muestra el contenido del volcado de memoria en una determinada ubicación. Se pueden observar indicios de una página web que se ha visitado.

[6] *Transmission Control Protocol* (TCP) y *User Datagram Protocol* (UDP) son protocolos de comunicación de la suite de Protocolos de Internet. Los *sockets* son mecanismos de los Sistemas Operativos para implementar conexiones.

Cualquiera sea la pericia a realizar, se deben considerar todos los aspectos detallados anteriormente en este capítulo. Los datos contenidos en un volcado de memoria se almacenan de diferentes maneras y pueden formar parte o no de distintas estructuras. Dichos datos pueden representar punteros a diferentes ubicaciones de la memoria principal, fechas, valores enteros o cadenas de texto, incluso es posible que estén almacenados en sistema *little-endian o big-endian*[7] con un sistema que puede ser distinto al que utiliza nativamente la plataforma.

Por todo esto es necesario saber cómo interpretarlos y qué herramientas utilizar como soporte de análisis. El siguiente ejemplo, muestra cómo se vería el contenido con un editor de texto tradicional y los datos que brinda un editor hexadecimal. Luego, cómo es posible realizar interpretaciones y traducciones de esos datos en formatos legibles para el analista forense gracias al uso de herramientas de soporte para el análisis.

En la Figura 4 se muestra el contenido del volcado de memoria visto a través de un editor hexadecimal[8] con comentarios sobre estructuras o datos identificados con BIP-M *framework*:

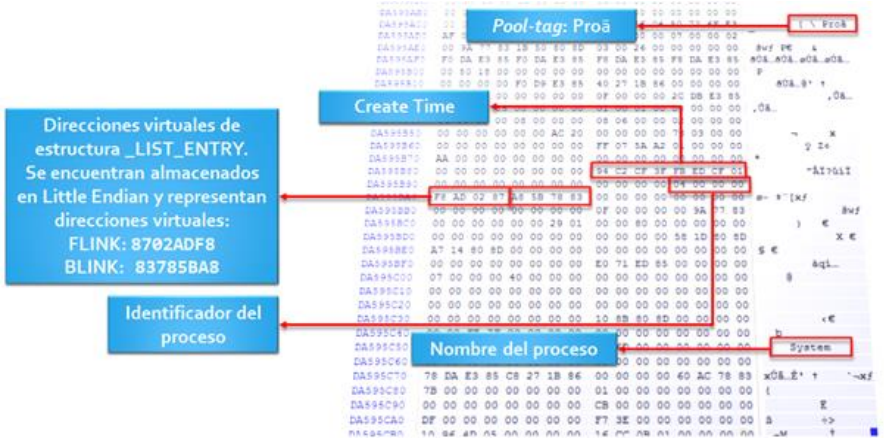

Fig. 4. Contenido del volcado de memoria visto con un editor hexadecimal.

En la Figura 5 se muestra el mismo contenido traducido y obtenido por BIP-M *framework*:

---

[7] El término inglés *endianness* designa el formato en el que se almacenan los datos de más de un byte en un ordenador. Usando este criterio el sistema *big-endian* adoptado por Motorola entre otros, consiste en representar los bytes en el orden "natural": así el valor hexadecimal 0x4A3B2C1D se codificaría en memoria en la secuencia {4A, 3B, 2C, 1D}. En el sistema *little-endian* adoptado por Intel, entre otros, el mismo valor se codificaría como {1D, 2C, 3B, 4A}.

[8] Un editor hexadecimal permite ver el contenido de un archivo tal cual es, mientras que un editor de texto convencional interpreta el contenido y lo representa como caracteres.

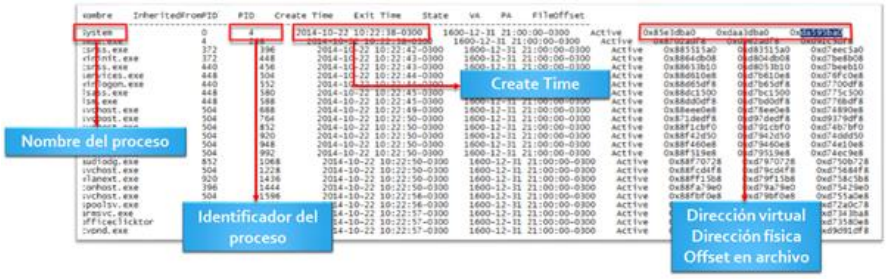

Fig. 5. Contenido del volcado de memoria obtenido a través de BIP-M *framework*. En particular, se muestra la lista de procesos activos en memoria en el momento de la captura del volcado.

7 Conclusiones

El análisis forense de la memoria principal puede aportar información de gran relevancia para una investigación judicial. En ciertas ocasiones incluso, los objetos en memoria son la única fuente para obtener determinados datos de interés, lo que aumenta la importancia de realizar este tipo de análisis. Todos los actores involucrados son claves, desde el principio aquellos que participan en la adquisición del volcado de memoria hasta aquellos encargados realizar el análisis del mismo.

La memoria principal alberga estructuras variadas, por lo que el analista forense debe tener un conocimiento profundo de lo que puede encontrarse y enfocarse en las estructuras importantes, evidente u ocultas, para el hallazgo de información y pruebas de la actividad del usuario o de la presencia de *malware* en el equipo. El hallazgo de un software malintencionado puede ser clave en una investigación, revelando al verdadero culpable detrás de un hecho. Por las características complejas de las amenazas informáticas modernas, la mejor forma de encontrar *malware* es el análisis de la memoria.

El Proyecto BIP-M, realizado en el marco de un proyecto final de graduación en la Universidad FASTA, nutrió al Grupo de Investigación en Sistemas Operativos e Informática Forense de conocimiento y experticia únicos en el tema de análisis de memoria. Además, se desarrolló un *framework* de análisis competitivo que se utiliza internamente para validación y pruebas, permitiendo comprender a mayor profundidad el contenido de un volcado de memoria y adaptarse a diferentes sistemas operativos.

La adopción de este tipo de análisis técnico en las investigaciones y las causas judiciales, enmarcadas en protocolos y procedimientos adecuados, le brinda a la justicia una nueva y poderosa herramienta capaz de obtener información y evidencia, que, con las técnicas clásicas, es casi imposible obtener.

Referencias

1. B. Kaplan, “RAM is Key: Extracting Disk Encryption Keys From Volatile Memory”, 2007
2. M. Burdach, “Finding Digital Evidence In Physical Memory”, Black Hat Federal, 2008
3. M. Sikorski, A. Honig, “Practical Malware Analysis”, 2013
4. M. Russinovich, D. Solomon, A. Ionescu, “Windows Internals 5”, 2009
5. M. Russinovich, D. Solomon, A. Ionescu, “Windows Internals 6”, 2012
6. A. Di Iorio, H. Curti, F. Greco, A. Podestá, M. Castellote, J. Iturriaga, S. Trigo, B. Constanzo, G. Ruiz de Angeli, S. Lamperti, “Construyendo una Guía Integral de Informática Forense”, CONAIISI 2015
7. A. Schuster, “Searching for processes and threads in Microsoft”, 2006.
8. M. H. Ligh, A. Case, J. Levy, A. Walters, “The Art of Memory Forensics”, 2014